RANCANGAN APLIKASI PENGAMANAN DATA DENGAN ALGORITMA *ADVANCED ENCYPTION STANDARD (AES)*

Angga Aditya Permana[1], Desi Nurnaningsih[2]

[1,2]Universitas Muhammadiyah Tangerang / Fakultas Teknik, Program Studi Informatika
Jl. Perintis Kemerdekaan 1/33 Cikokol Kota Tangerang TLP. 021 55793251
e-mail: angga.permana@umt.ac.id[1], desi.nurnaningsih@umt.ac.id[2]

ABSTRAK

Data sangatlah berharga bagi semua pengguna komputer, belakangan ini kriptografi menjadi metode yang digunakan dalam mengamankan data. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika dalam mengamankan suatu informasi atau pesan asli (*Plainteks*) menjadi sebuah teks tersembunyi (*Chiperteks*) dan kemudian di ubah menjadi pesan asli kembali. Kriptografi mempunyai tiga unsur penting yaitu pembangkitan kunci, enkripsi dan deskipsi. Dalam kriptografi di kenal algoritma block chiper yang didalamnya terdapat AES (*Anvanced Encyption Standard*) merupakan bagian dari *Modern Symmetric Key Cipher*, algoritma ini menggunakan kunci yang sama pada saat proses enkripsi dan deskripsi sehingga data yang kita miliki akan sulit dimengerti maknanya. Teknik algoritma tersebut digunakan untuk mengkonversi data dalam bentuk kode-kode tertentu, untuk tujuan agar informasi yang tersimpan tidak bisa di baca siapa pun kecuali orang-orang yang berhak. Oleh karena itu, sistem keamanan data sangat di perlukan untuk menjaga kerahasian informasi agar tetap terjaga.

Kata kunci: *Algoritma, Kriptografi, AES, Enkripsi, Deskripsi*

ABSTRACT

Data is a valueable for all computer users, cryptography is the one of method used to securing data. Cryptography is the study of mathematical techniques in securing an information or original message (Plainteks) into a hidden text (Chiperteks) and then converted into the original message again. Cryptography has three important elements, first is key generation, second is encryption and latetly is description. In cryptography known as cipher block algorithms in which AES (Advanced Encyption Standard) is part of Modern Symmetric Key Cipher, this algorithm uses the same key during the encryption and description process so that the data we have will be difficult to understand. The algorithm technique is used to convert data in the form of certain codes, so that the information stored cannot be read by anyone except those who are entitled. Therefore, the data security system is very necessary to maintain the confidentiality of information.

Keywords: *Algoritma, Kriptografi, AES, Enkripsi, Deskripsi*

http://dx.doi.org/10.15408/jti.v11i2.7811

I. PENDAHULUAN

Seiring dengan kemajuan teknologi informasi maka sangat di perlukan sebuah keamanan data terhadap kerahasiaan informasi yang saling di pertukarkan melalui jaringan internet, apa lagi jika data tersebut dalam suatu jaringan komputer yang terhubung/terkoneksi dengan jaringan lain. Hal tersebut tentu saja menimbulkan resiko bila informasi yang sensitif dan berharga tersebut di akses oleh orang yang tidak bertangung jawab. Yang mana jika hal tersebut sampai terjadi, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim pesan, maupun organisasinya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar.

Pada penelitian sebelumnya telah dirancang aplikasi penyisipan teks pada gambar dengan algoritma blowfish dan least significant bit [7] namun agoritma tersebut sangat mudah di tebak oleh orang yang tidak bertanggung jawab. Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *AES* untuk proses enkripsi dan deskripsi data. Kriptografi telah menjadi suatu bagian yang tidak dapat di pisahkan dari sistem keamanan jaringan, salah satu metode enkripsi data adalah *Advanced Encryption Standard* (*AES*). .

II. TINJAUAN PUSTAKA

2.1 Definisi Perancangan

Pada saat hendak membuat sebuah sistem yang akan digunakan oleh pengguna (*user)*, setiap pengembang aplikasi diharuskan membuat sebuah rancangan dari sistem yang ingin dibuat. Rancangan ini bertujuan untuk memberi gambaran umum dari sistem yang akan berjalan kepada setiap pengguna. Perancangan adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan pengguna (*user*).

2.2. Pengamanan Data

Secara umum data dikategorikan menjadi dua, yaitu data yang bersifat rahasia dan data yang tidak bersifat rahasia. Data yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Yang sangat perlu diperhatikan adalah data yang bersifat rahasia, dimana setiap informasi yang ada didalamnya akan sangat berharga bagi pihak yang membutuhkan karena data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasanya dilakukan berbagai cara yang tidak sah.

2.3. Algoritma

"Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematik dan logis". Kata logis merupakan kata kunci dalam algoritma. Langka-langka dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar [5].

2.4. Definisi Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata "*graphy*" di dalam "*cryptography*" itu sendiri sudah menyiratkan sebuah seni [5].

Untuk dapat menjalankan dengan baik pada proses kriptografi haruslah terdapat emapat elemen utama didalamnya, yang paling berkait satu sama lain. Yaitu :

1. *Plain Text*
   Merupakan sebagai pesan awal atau pesan asli yang di kirim pada proses komunikasi. *Plain Text* inilah yang kemudian di enkripsi dan di deskripsi.
2. *Cipher Text*
   Merupaka pesan yang tersembunyi, yaitu pesan asli (*Pain Text*) yang telah di enkripsi dapa proses kriptografi. *Cipher Text* ini dapat diubah kembali kebentuk aslinya (*Pain Text*) memanfaatkan *Key* yang telah di sediakan.
3. *Cryptography Key*
   Merupakan kunci yang di gunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidaka dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.

4. *Encryption Decryption Algorithm*
   Komponen terakhir yang juga samapentingnya dalam proses kriptografi adalah algoritma yang di gunakan untuk enkripsi dan dekripsi.

Jenis serangan berdasarkan cara dan posisi seseorang untuk mendapatkan pesan-pesan dalam jaringan, yaitu :

1. *Sniffing*
   Sniffing berarti 'mengendus', dalam hal ini yang diendus merupakan pesan (baik yang belum ataupun yang sudah di enkripsi) dalam suatu saluran komunikasi. Hal ini umum terjadi pada saluran publik yang tidak aman yang mengakbatkan sang pengendus dapat merekam pembicaraan yang terjadi.
2. *Replay Attack*
   Replay Attack adalah serangan jaringan dimana penyerang menyadap percakapan antara pengirim dan penerima, serta mengambil informasi autentik dengan berbagi kunci. Penyerang kemudian menghubungi penerima dengan kunci itu sebagai bukti identitas dan keaslian untuk menipu penerima.
   Misalkan Yudhit mencuri informasi yang dikirimkan Bagas ke Salsa selanjutnya mengubah pesan tersebut sebelum dikirim kembali ke Salsa seolah pesan tersebut asli dari Bagas.
3. *Spoofing*
   Spoofing adalah teknik untuk memperoleh akses yang tidak sah ke suatu komputer atau informasi dimana penyerang berhubungan dengan pengguna dengan berpura-pura sebagai host yang dapat dipercaya. *Spoofing* biasanya dilakukan oleh seorang *hakcer/craker.*
   Sebagai contoh, Seorang penyerang (misalnya Yudhit) bisa menyamar menjadi Bagas. Semua orang dibuat percaya bahwa Yudhit adalah Bagas. Penyerang berusaha meyakinkan kepada pihak-pihak lain bahwa tidak ada yang salah dengan komunikasi yang dilakukan. Padahal komunikasi itu dilakukan dengan penipu/penyerang.
4. *Man-in-the-middle*
   Jika s*poofing* kadang hanya menipu satu pihak, dalam skenario ini saat Bagas hendak berkomunikasi dengan Salsa, Yudhit dimata Bagas seolah-olah menjadi Salsa. Ydhit juga dapat menipu Salsa sehingga ia seolah-olah adalah Bagas. Yudhit dapat berkuasa penuh atas jalur komunikasi dan bisa membuat berita fitnah.

2.5. Definisi *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunkan untuk mengamakan data. Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut ciphertext; sebaliknya dekripsi adalah merubah ciphertext data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekripsi data.

Secara umum metode yang digunakan dalam pemrosesan terbagi dua, yaitu

1. Enkripsi
   Enkripsi adalah proses penyandian *plaintext* menjadi *ciphertext*, atau pengubahan data menjadi bentuk rahasia.
   Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes, ShiftRows, Mixcolumns,* dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah di*copy*kan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*.
2. Dekripsi
   Dekripsi adalah proses memperoleh kembali *plaintext* menjadi *ciphertext,* atau proses pengubahan kembali data yang berbentuk rahasia menjadi semula. Transformasi byte yang digunakan pada *invers cipher* adalah *InvShiftRows, InvSubBytes, InvMixColumns*, dan *AddRoundKey*.

III. METODOLOGI

Studi Pustaka *(literature study)* dilaksanakan dengan cara mengumpulkan dan mempelajari segala macam informasi yang berhubungan dengan kriptografi, algoritma *AES* dan segala hal yang berhubungan dengan model pemrogramannya, bisa melalui buku dan internet.

Metode Pengembangan Sistem

1. Analisa Kebutuhan
   Perkembangan teknologi informasi dan komunikasi sangat cepat dan pesat, hal ini yang menyebabkan munculnya kemajuan teknologi informasi. Maka sangat di perlukan sebuah keamanan data terhadap kerahasiaan informasi yang saling di pertukarkan melalui jaringan internet. Pembuatan *software* aplikasi sistem keamanan data menggunakan Microsoft Visual Studio 2010. Microsoft Visual Studio 2010 mengatasi semua masalah pengembangan apliakasi berbasis Windows dan memiliki fasilitas penanganan Bug yang hebat dan Real Time Backroud Compiler. Hal ini akan memberi kontribusi yang kuat dalam sistem enkripsi dan deskripsi. Dan harus menganalisa terlebih dahulu proses enkripsi dan dekripsi pada algoritma AES Rijndael.
2. Desain Sistem
   Pada tahap ini dilaksanakan perancangan Aplikasi Sistem yang akan dibuat berdasarkan hasil Studi Pustaka *(study literature)* yang ada. Perancangan Aplikasi sistem ini meliputi desain menu utama yang terdiri dari enkripsi dekripsi file dan teks serta pemrograman. Perencanaan penggunaan bahasa pemrograman Microsoft Visual Studio 2010.
3. Implementasi
   Dalam tahap ini, dilakukan implementasi berdasarkan studi pustaka dan rancangan yang telah dibuat pada tahap sebelumnya, dalam bentuk program.
4. Uji Coba dan Evaluasi
   Merupakan kegiatan mengevaluasi dan memperhalus analisis kebutuhan calon pengguna. Program yang sudah diuji coba program untuk mencari masalah yang mungkin timbul dan disempurnakan kemudian dievaluasi kebenaran dan kemampuannya terhadap sistem serta mengevaluasi jalannya program.
5. Analisa Hasil Uji Coba
   Pada tahap ini dihasilkan uji coba dan beberapa revisi, jika terjadi kekurangan dan kesalahan terhadap perangkat lunak yang telah selesai dibuat dan diharapkan sistem aplikasi tersebut mengalami segala macam uji coba sehingga menghasilkan *output* yang diharapkan.

IV. ANALISA DAN PEMBAHASAN

4.1 Analisa Kebutuhan *Software*

Perkembangan teknologi informasi dan komunikasi sangat cepat dan pesat, hal ini yang menyebabkan munculnya kemajuan teknologi informasi. Maka sangat di perlukan sebuah keamanan data terhadap kerahasiaan informasi yang saling di pertukarkan melalui jaringan internet.

4.2. Identifikasi Permasalahan

Seiring dengan kemajuan teknologi informasi yang semakin pesat maka sangat di perlukan sebuah keamanan data terhadap kerahasiaan informasi. Sangat banyak informasi yang sensitif dan berharga tersebut di akses oleh orang yang tidak bertangung jawab, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim pesan, maupun penerimanya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar. Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *AES (Advanced Encryption Standard)* untuk proses enkripsi dan deskripsi data.

4.3. Analisa Kebutuhan

Dalam kriptografi dikenal algoritma *block chiper* yang di dalamnya terdapat AES (*Anvanced Encyption Standard*) merupakan bagian dari *Modern Symmetric Key Cipher*, algoritma ini menggunakan kunci yang sama pada saat proses enkripsi dan deskripsi sehingga data yang kita miliki akan sulit dimengerti maknanya.

Teknik algoritma tersebut digunakan untuk mengkonversi data dalam bentuk kode-kode tertentu, untuk tujuan agar informasi yang tersimpan tidak bisa di baca siapa pun kecuali orang-orang yang berhak. Oleh karena itu, sistem keamanan data sangat di perlukan

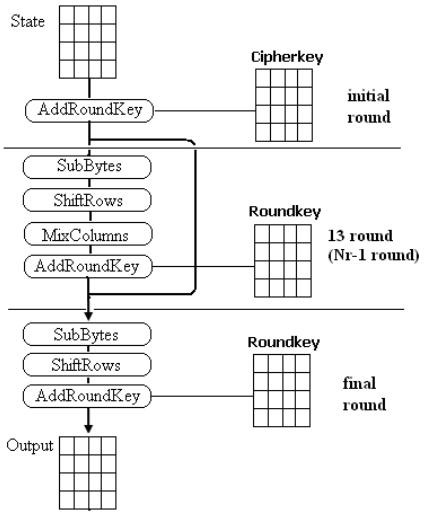
untuk menjaga kerahasian informasi agar tetap terjaga.

Pembuatan *software* aplikasi sistem keamanan data menggunakan Microsoft Visual Studio 2010. Microsoft Visual Studio 2010 mengatasi semua masalah pengembangan apliakasi berbasis Windows dan memiliki fasilitas penanganan Bug yang hebat dan Real Time Backroud Compiler. Hal ini akan memberi kontribusi yang kuat dalam sistem enkripsi dan deskripsi.

4.4. Desain Algoritma Pada Kasus

4.4.1 Proses Enkripsi

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah di*copy*kan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar di bawah ini :



Gambar 1. Diagram enkripsi AES [5]

Garis besar algoritma AES Rijndael yang beroprasi pada blok 128-bit dengan kunci 128-bit (diluar proses pembangkitan *roundkey*) adalah sebagai berikut :

1. *AddRoundKey*, melakukan XOR antara awal (*plantext*) dengan *cipher key*.
2. Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran adalah :
   a. *SubBytes* adalah subsitusi *byte* menggunakan tabel subsitusi (*S-Box*).
   b. *ShiftRows* adalah pergeseran baris-baris *array state* secara *wrapping.*
   c. *MixColumns* adalah mengacak data di masing-masing kolom *array state.*
   d. *AddRoundKey* adalah melakukan XOR antara state sekarang *round key.*
3. *Final round,* proses untuk putaran terakhir :
   a. *SubBytes*
   b. *ShiftRows*
   c. *AddRoundKey*

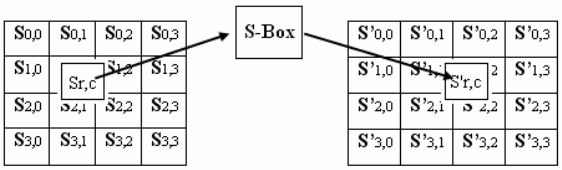
Langkah kerja enkripsi adalah sebagai berikut:

a. Transformasi *SubBytes*
   SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam Tabel 1.

Tabel 1. S-Box Rijndael [5]

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Untuk setiap *byte* pada *array state*, misalkan *S*[*r*, *c*] = *xy*, yang dalam hal ini *xy* adalah *digit* heksadesimal dari nilai *S*[*r*, *c*], maka nilai substitusinya, dinyatakan dengan *S*'[*r*, *c*], adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris *x* dengan kolom *y*. Gambar 2 mengilustrasikan pengaruh pemetaan *byte* pada setiap *byte* dalam *state*.

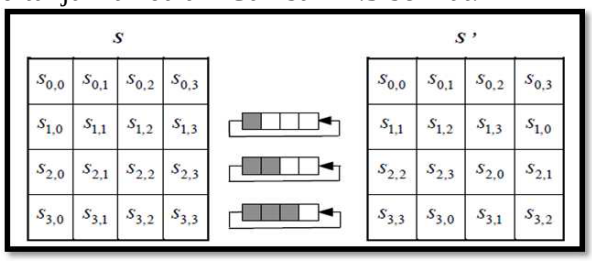


Gambar 2. Pengaruh pemetaan pada setiapb*Byte* dalam *state* [3]

b. *Shiftrows*

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran *bit* dimana *bit* paling kiri akan dipindahkan menjadi *bit* paling kanan

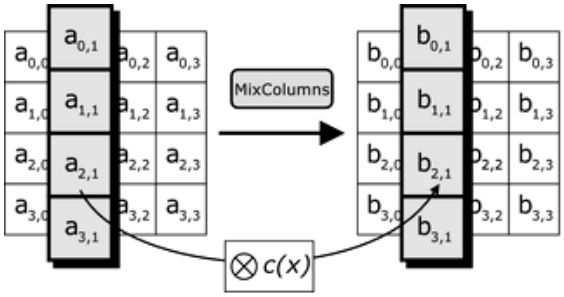
(rotasi *bit*). Proses pergeseran *Shiftrow* ditunjukkan dalam Gambar III.3 berikut:



Gambar 3. Proses *shifrows* [5]

c. *MixColumns*

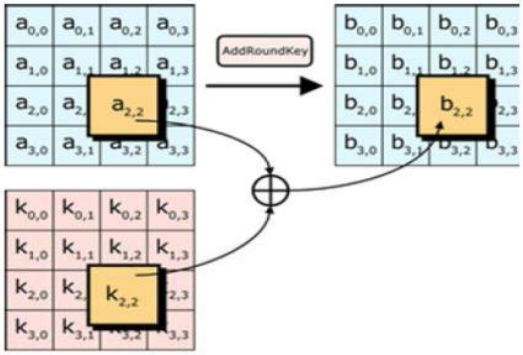
MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi *mixcolumns* dapat dilihat pada perkalian matriks berikut ini:



Gambar 4. Proses *mixcolumns* [5]

d. *AddRoundKey*

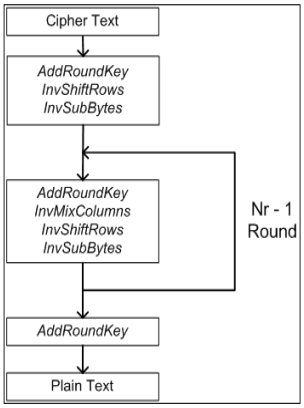
AddRoundKey: melakukan *XOR* antara *state* sekarang dengan *round key*.



Gambar 5. Proses *addroundKey* [5]

4.4.2 Proses Dekripsi

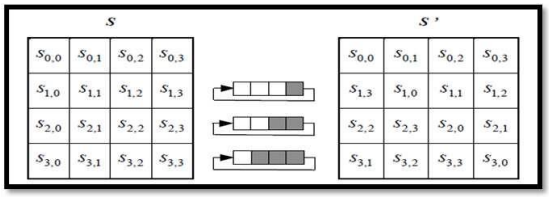
Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada skema berikut ini :



Gambar 6. Proses dekripsi [5]

a. InvShiftRows

InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi *InvShiftRows* terdapat pada Gambar 7:



Gambar 7. Proses *invshiftrows* [5]

b. InvSubBytes

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse S-Box*.

Hex		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	E3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	17	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Gambar 8. Proses *invsubbytes* [5]

c. InvMixColumns

Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan :

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E\ 0B\ 0D\ \theta \\ \theta\ 0E\ 0B\ 0D \\ 0D\ \theta\ 0E\ 0B \\ 0B\ 0D\ \theta\ 0E \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Hasil dari perkalian matriks adalah

$$S'_{0,c} = (\{0E\} \bullet S_{0,c}) \oplus (\{0B\} \bullet S_{1,c}) \oplus (\{0D\} \bullet S_{2,c}) \oplus (\{\theta\} \bullet S_{3,c})$$
$$S'_{1,c} = (\{\theta\} \bullet S_{0,c}) \oplus (\{0E\} \bullet S_{1,c}) \oplus (\{0B\} \bullet S_{2,c}) \oplus (\{0D\} \bullet S_{3,c})$$
$$S'_{2,c} = (\{0D\} \bullet S_{0,c}) \oplus (\{\theta\} \bullet S_{1,c}) \oplus (\{0E\} \bullet S_{2,c}) \oplus (\{0B\} \bullet S_{3,c})$$
$$S'_{3,c} = (\{0B\} \bullet S_{0,c}) \oplus (\{0D\} \bullet S_{1,c}) \oplus (\{\theta\} \bullet S_{2,c}) \oplus (\{0E\} \bullet S_{3,c})$$

d. Inverse AddRounKey

Transormasi *Inverse AddRounKey* tidak berbeda dengan transformasi *AddRoundkey* karna dalam transformasi ini hanya dilakukan operasi penambahan sederhana dengan operasi *bitwise* XOR.

Software Architecture

```
case"Encryption":
    {
        try
        {
            string password=textBox3.Text;
            UnicodeEncoding UE= new UnicodeEncoding();
            byte[]key=UE.GetBytes(password);
                string cryptedfile= textBox2.Text;
            FileStream FsCrypt= new FileStream(cryptedfile, FileMode.Create);
            RijndaelManaged RMCrypt= new RijndaelManaged();
            CryptoStream CS = new CryptoStream(FsCrypt,RMCrypt.CreateEncryptor(key,key),CryptoStreamMode.Write);
            FileStream FSIN = new FileStream(textBox1.Text,FileMode.Open);
            int data;
            while ((data=FSIN.ReadByte())!=-1)
                CS.WriteByte((byte)data);
            FSIN.Close();
            CS.Close();
            FsCrypt.Close();

            label2.Text = "Done";
            MessageBox.Show("the encryption operation is done correctly");


        }
        catch(Exception ex)
        {
            label2.Text="the encryption opration failed";
            MessageBox.Show("Error" + ex);
        }
        break;
```

Gambar 9. Proses Enkripsi

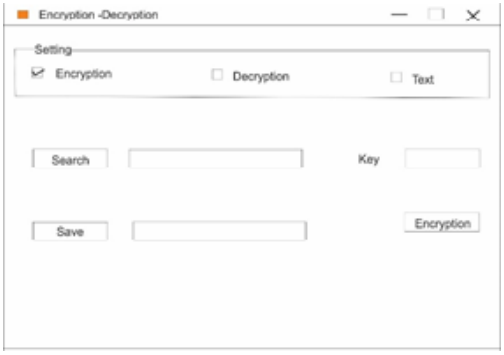
```
    }
    case "Decryption":
            {
                try
                {
                    string password=textBox3.Text;
                UnicodeEncoding UE= new UnicodeEncoding();
                byte[]key=UE.GetBytes(password);
                    string cryptedfile= textBox1.Text;
                FileStream FsCrypt= new FileStream(cryptedfile, FileMode.Open);
                RijndaelManaged RMCrypt= new RijndaelManaged();
                CryptoStream CS = new CryptoStream(FsCrypt,RMCrypt.CreateDecryptor(key,key),CryptoStreamMode.Read);
                FileStream FSout = new FileStream(textBox2.Text,FileMode.Create);
                int data;
                while ((data=CS.ReadByte()) !=-1)
                    FSout.WriteByte((byte)data);
                FSout.Close();
                CS.Close();
                FsCrypt.Close();
                label2.Text = "done";
                MessageBox.Show("the decryption operation is done correctly");

                }
                catch(Exception ex)
                {
                    label2.Text="operation failed";
                    MessageBox.Show("Error"+ex);
                }

                break;
            }
```

Gambar 10. Proses Dekripsi

4.4.3 *User Interface*

Pada Perancangan aplikasi ini terdapat dua keguanan, yaitu untuk enkripsi dan dekripsi file serta untuk enkripsi dan dekripsi *text*. Tampialan awal aplikasi yang dibuat seperti ini:

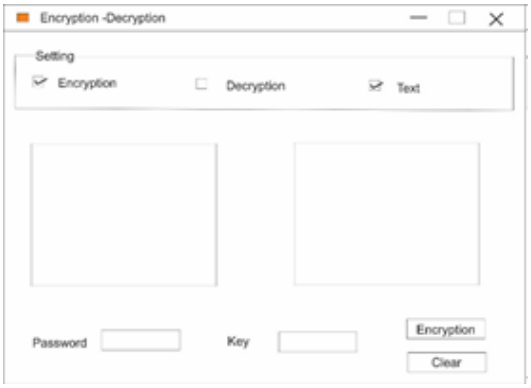
1. Perancangan Enkripsi dan Dekripsi File



Gambar 11. Perancangan enkripsi dan dekripsi file

Untuk mengenkripsi atau mengdekripsi data file, pilih *Encryption* atau *Decryption* dalam kotak *Setting*. Kemudian cari (*Search*) file data yang dituju dan simpan (*Save*) di folder tertentu, masukan kunci (*Key*) 8 karakter dan jalankan program. Hasil akan langsung tersimpan di folder yang telah dipilih sebelumnya.

2. Perancangan Enkripsi dan Dekripsi Data



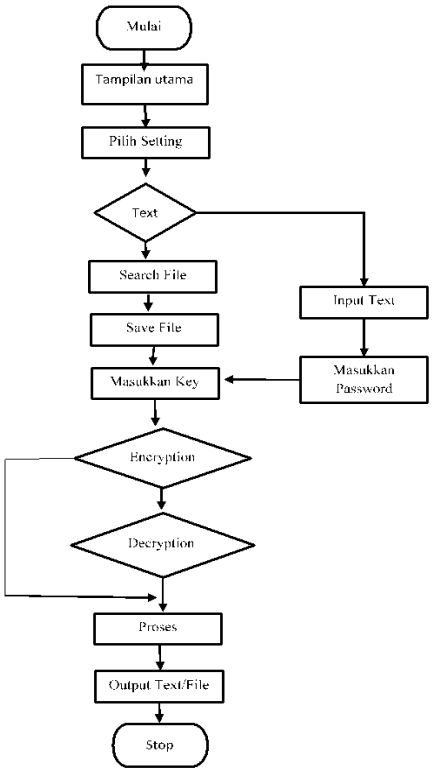
Gambar 12. Perancangan enkripsi dekripsi teks

Pengubahan *text* menjadi kode-kode atau sebaliknya. Sebelumnya pilih *Text* pada kotak *Setting*. Tuliskan *text* di kotak kiri, masukan *password* dan *key* 8 karakter kemudian jalankan program enkripsi. Dan hasil enkripsi akan muncul dikotak kanan. Pengubahan dari dekripsi ke enkripsi tuliskan kode-kode yang berbentuk kalimat yang telah dienkripsi ke dalam kotak kanan, pilih *setting* dekripsi masukan *password* dan *key* 8 karakter, dan hasilnya dekripsi akan muncul di kotak kiri.

4.4.4 *Testing*

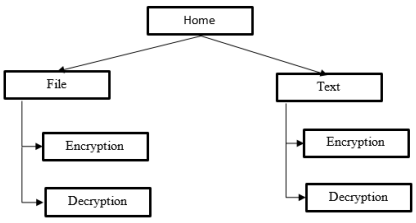
Perancangan Struktur Program

Adapun struktur program dalam aplikasi keaman data enkripsi dan dekripsi adalah sebagai berikut:



Gambar 13. Struktur program menu utama

4.4.5 Perancangan Aplikasi Enkripsi Deskripsi

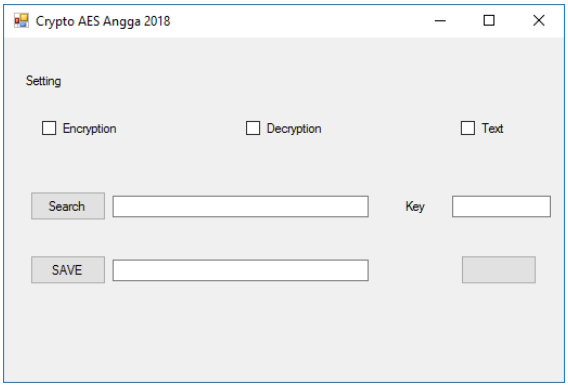


Gambar 14. Perancangan aplikasi enkripsi dekripsi

4.4.6 Implementasi

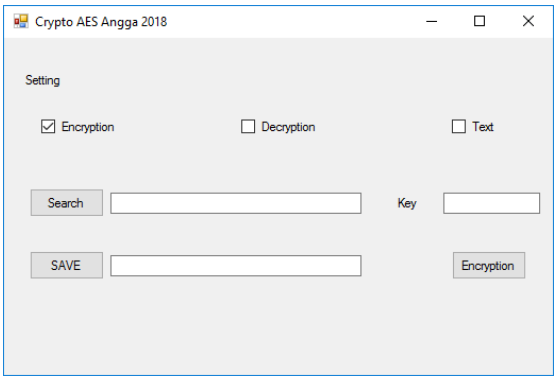
Implementasi algorotma AES (*Advanced Encryption Standard*) Rijdael dibuat Menggunakan Microsoft Visual Studio 2010. Aplikasi yang terdiri dari enkripsi dekripsi *text* dan enkripsi dekripsi file.

4.4.7 Implementasi *Encryption File*

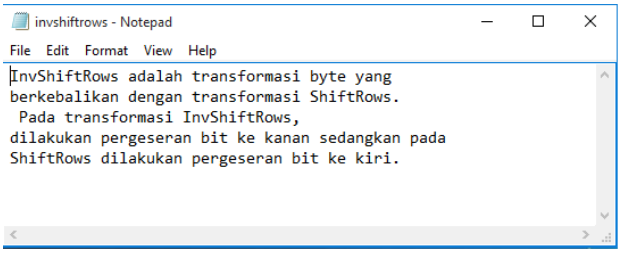


Gambar 15. Tampilan utama aplikasi enkripsi dekripsi

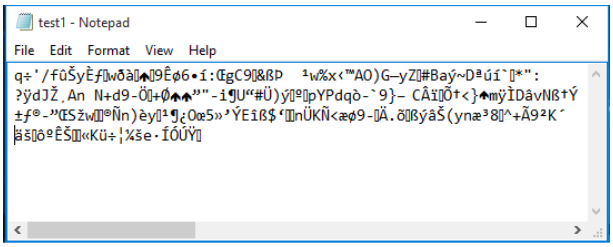
Pada tampilan utama aplikasi terdapat *setting* yang terdiri dari *encryption*, *decryption*, dan *text*. Serta difungsikan langsung untuk enkripsi atau dekripsi file.



Gambar 16. Pilih *setting encryption*



Gambar 17. File yang akan dienkripsi



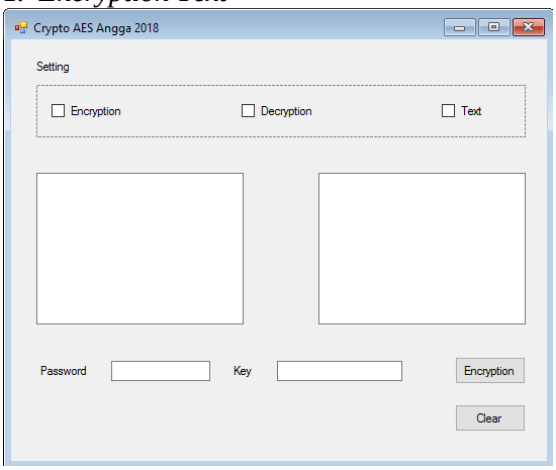
Gambar 18. Hasil enkripsi

Pada enkripsi dekripsi file, file yang akan dienkripsi ialah file-file yang terdapat

dalam sistem komputer seperti file Microsoft Office, file MP3, file PDF, dan file gambar. File yang telah dienkripsi akan berekstensi *aes* dan untuk melihat hasil dari enkripsi file tersebut dapat dilihat menggunakan *notepad.*

4.4.8 Implementasi *Text*

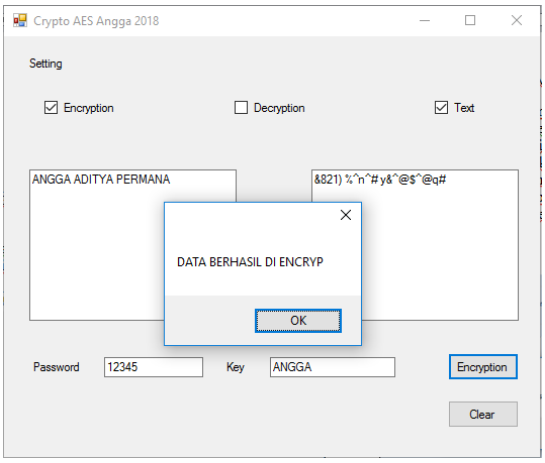
1. *Encryption Text*



Gambar 19. Mengenkripsi *text*

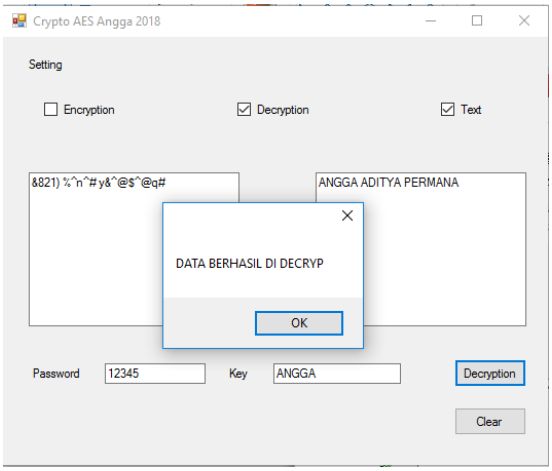
Pada *setting* terdapat pilihan *text* untuk mengenkripsi dekripsi *text*, dan tampilan aplikasi kemudian berubah seperti pada Gambar 20.

Untuk enkripsi teks, tuliskan teks pada kolom kiri dan hasilnya akan muncul dikolom kanan. Dan jika mendekripsi teks, menginput text enkripsi pada kolom kanan kemudian hasil teks dekripsi akan muncul di kolom kiri.



Gambar 20. Enkripsi *text* berhasil

4.4.9 *Decryption Text*



Gambar 21. Dekripsi *text*

Pada enkripsi file dan teks jika terjadi perubahan kunci, maka hasilnya tidak akan terbaca dan dingggap file tersebut rusak. Jika terjadi perbedaan, maka disimpulkan ada modifikasi terhadap isi pesan. Aplikasi ini didasarkan pada kenyataan bahwa perubahan kunci akanmenyebabkan perubahan data pada saat akan dikembalikan pada bentuk semula atau didekripsi dan hasilnya akan berbeda dengan aslinya (plainteks). Dengan kata lain, *Rijndael* sangat peka terhadap perubahan sekecil apa pun pada data masukan.

V. PENUTUP

Pembuatan aplikasi sistem pengamanan data enkripsi dekripsi file dan teks menggunakan Microsoft Visual Studio 2010 mendukung perkembangan zaman yang semakin canggih. Pemilihan algoritma algoritma AES: Rijndael, karena algoritma ini merupakan algoritma yang cukup sulit dipecahkan saat ini, sebab belum ada serangan atau pemecahan yang belum mampu secara analisis matematika dengan efektif dan efesien dengan alasan pola yang dibentuk cukup acak. AES: Rijndael memiliki keunggulan karena memiliki daya memori dan kecepatan komputasi dalam pengoprasian. Pengoprasian yang tidak memakan memori yang telalu besar ini banyak diminati pasar karena kebutuhan efesiensi waktu yang relatif cepat.

Diharapkan kepada pembaca yang melakukan proses enkripsi dan dekripsi file ataupun teks untuk menggunkan kunci yang lebih variatif atara huruf dan angka ataupun

karakter yang akan lebih menyulitkan pemecahan cipherteks oleh kriptanalis.

DAFTAR PUSTAKA

[1] Agus Eka Pratama, I Putu. 2014. Handbook Jaringan Komputer. Bandung: Informatika.

[2] Anhar. 2010. Cara Mudah Mengamankan Data Komputer dan Laptop. Jakarta: Media Kita.

[3] Ncbnzx Fachrurozi, Muhammad Farid. 2006. Enkripsi Pesan Rahasia Menggunakan Algoritma (*Advandes Encryption Standard*) AES: Rijndael. Jakarta: Universitas Islam Negri Syarif Hidayatullah Jakarta.

[4] Ibrahim, Rohmat Nur. 2012. Kriptografi Algoritma DES, AES/Rijndael, Blowfish Untuk Keamanan Citra DigitalDengan Menggunakan Metode Discrete Wavelete Transformasion (DWT). ISSN: 2442-4943. Bandung: Jurnal Computech and Bisnis, Vol. 6, No.2, Desember 2012, 82-95.

[5] Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Informatika Bandung.

[6] Permana AA, 2016. Rancangan Sistem Informasi Simpan Pinjam Pada Koperasi Guru Dan Pegawai SMP Negeri 45 Jakarta. ISSN: 2088-0154. Vol 8, No 1 - 2016

[7] Permana AA, 2017. Aplikasi penyisipan teks pada gambar dengan algoritma blowfish dan least significant bit. ISSN: 2549-0710. Tangerang: Jurnal Informatika, Vol 1, No 1 Januari 2017, 11-17.

[8] Primartha, Rifkie. 2013. Penerapan Enkripsi dan Dekripsi File Menggunakan Algoritma *Advens Encryption Standard* (AES). ISSN: 2301-8488. Palembang: *Journal of Research in Computer Science and Applications* – Vol. 2, No. 1, Januari 2013: 13-18.

[9] Rahmayunita, Isnawaty, Sutardi. 2015. Penyadapan SMS dan GPS Berbasis Android Menggunakan Algoritma *Advanced Encryption Standard* (AES). ISSN: 2460-1446. Kendari. SemanTIK, Vol. 1, No.2, Juli-Desember 2015, pp. 11-22.

[10] Riadi, Muchlisin. 2014. Pengertian, Sejarah dan Jenis Kriptografi. Diambil dari: http://www.kajianpustaka.com/2014/01/pengertian-sejarah-dan-jenis-kriptografi.html?m=1 .

[11] Satria, Eko. 2009. Studi Algoritma Rijndael dalam Sistem Keamanan Data. Sumatra Utara: USU Repository.

[12] Setyaningsih, Emy. 2015. Kriptografi dan Implementasinya Menggunakan MATLAB. Yogyakarta: Andi

[13] Sianturi, Fricles Ariwisanto. 2013. Perancangan Aplikasi Pengamanan Data Dengan Kriptografi *Advances Encryption Standard* (AES). ISSN: 2301-9425. Medan: Pelita Informatika Budi Darma, Vol. 4, No. 1, Agustus 2013: 42-46.

[14] Sutejo, Budi dan Michael. 2004. Algoritma dan Teknik Pemprograman. Yogyakarta: Andi